Fitting 3d models on central catadioptric images

Eric Marchand, François Chaumette

IEEE Int. Conf on Robotics and Automation, ICRA'07
Roma, Italia, April 2007

Fitting 3D Models on Central Catadioptric Images

Eric Marchand, François Chaumette

***Abstract*— Increasing the field of view of camera is an important issue practical in robot vision. One solution is to consider catadioptric camera that allows a 360^o field of view. In this paper we propose a 3D model tracking algorithm that allows a fast and reliable tracking of 3D objects within central catadioptric images. The proposed approach relies on the virtual visual servoing approach. All the modeling aspects have been reconsidered to consider the projection model. Results show the method to be robust and efficient.**

I. INTRODUCTION

Increasing the field of view of camera is an important issue in robot vision. Omnidirectional cameras have been recently widely studied in the literature. One solution to build such systems is to combine a mirror with a classical camera. Such systems are called catadioptric cameras. When a single projection center is sufficient to describe the projection in the image plane, these systems are referred as central catadioptric cameras.

Theoretical aspects of central catadioptric images are now well known [2], [12], and a lot of works have been done in the structure from motion area [13] and more recently in visual servoing [19]. Nevertheless few researches have been done to solve visual tracking problem which is a fundamental issue for the development of visual systems that consider such images. In [18] an SSD-based tracker allows the tracking of planar structure using an efficient second order minimization method. Closer to our problem is [3] where a model-based tracker based on a global non-linear minimization is presented. Although the goal is very similar, the modeling of the cost function as well as the minimization issue that we consider in this paper are different. In this paper we present a 3D model-based tracking, relying on the virtual visual servoing framework [5], [6]. Assuming small camera motions between two frames, tracking is reduced to a 3D camera localization problem.

When dealing with 3D camera localization or pose computation, most of the approaches proposed in the literature rely on a 3D registration issue. Considering perspective projection *full-scale non-linear optimization techniques* (e.g., [16], [7], [5], [6]) which consists of minimizing the error between the observation and the forward-projection of the model have proved to be very efficient. In this case, minimization is handled using numerical iterative algorithms such as Newton-Raphson or Levenberg-Marquardt [16]. These 2D-3D registration techniques rely on the use of a 3D model of the tracked objects. Considering catadioptric camera, similar approaches can be considered [3]. Nevertheless, a new projection model has to be considered which implies the definition of new visual features and of new Jacobians to be used in the minimization approach.

Authors are with INRIA, IRISA, Lagadic, F-35000 Rennes, France ; e-mail marchand@irisa.fr

In this paper, we first recall the virtual visual servoing approach in Section II. We then present in Section III the visual features used in the minimization process and we determine the analytical form of the corresponding Jacobian. In Section IV we describe the low level image processing method used in our tracker. Finally, experimental results are presented in Section V.

II. OVERVIEW OF THE TRACKING ALGORITHM

The fundamental principle of the virtual visual servoing approach [6] is to define the pose computation problem as the dual problem of 2D visual servoing [8], [15]. In visual servoing, the goal is to move a camera in order to observe an object at a given position in the image.

To illustrate the principle, consider the case of an object with various 3D features ${}^o\mathbf{P}$ (for instance, ${}^o\mathbf{P}$ are the 3D coordinates of some object points in the object frame). A virtual camera is defined, with same intrinsic parameters as the real camera and whose pose in the object frame is defined by the homogeneous matrix ${}^c\mathbf{M}_o$. The approach consists of estimating the real pose by minimizing the error Δ between the observed data $\mathbf{s}^*$ (the position of a set of features in the image) and the position $\mathbf{s}$ of the same features computed by forward-projection according to the current pose:

$$\Delta = \sum_{i=1}^{k} \left(\mathrm{pr}_\xi({}^c\mathbf{M}_o, {}^o\mathbf{P}_i) - s_i^* \right)^2, \quad (1)$$

where $\mathrm{pr}_\xi()$ is the projection model according to the camera intrinsic parameters ξ and where k is the number of considered features. Usually $\mathrm{pr}_\xi()$ is the perspective projection model [16], [7], [6] but any kind of projection model such as a projection model suitable for catadioptric cameras, can be considered as will be shown in the next section. We will see that all the modeling issue of the tracking have thus to be rewritten.

The virtual camera initially at ${}^{c_0}\mathbf{M}_o$ is then moved using a visual servoing control law in order to minimise the error Δ. At convergence, the virtual camera reaches the pose ${}^{c^*}\mathbf{M}_o$ which corresponds the real camera's pose (see Figure 1).

An important assumption is to consider that $\mathbf{s}^*$ is computed from the image with sufficient precision. When outliers are present in the measures, a robust estimation is required. M-estimators can be considered as a general form of maximum likelihood estimators [14]. Many functions have been

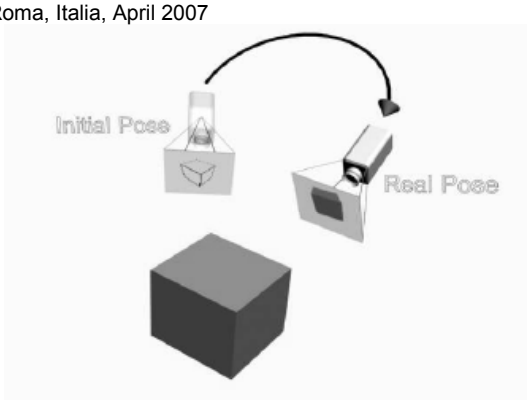

Fig. 1. Virtual visual servoing principle

proposed in the literature which allow uncertain measures to be less likely considered and in some cases completely rejected. In other words, the objective function is modified to reduce the sensitivity to outliers. The robust optimisation problem is then given by

$$\Delta_{\mathcal{R}} = \sum_{i=1}^{k} \rho\Big(s_i(\mathbf{r}) - s_i^*\Big), \tag{2}$$

where $\rho(u)$ is a robust function [14] that grows sub-quadratically and is monotonically non-decreasing with increasing $|u|$. Iteratively Re-weighted Least Squares (IRLS) is a common method of applying the M-estimator. It converts the M-estimation problem into an equivalent weighted least-squares problem. Thus, the error to be regulated to 0 is defined as

$$\mathbf{e} = \mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*), \tag{3}$$

where $\mathbf{D}$ is a diagonal weighting matrix given by $\mathbf{D} = \text{diag}(w_1, \ldots, w_k)$. Each element of $\mathbf{D}$ is a weight which is given to specify the confidence in each feature location. The computation of weights w_i is described in [6].

A simple control law that allows to move a virtual camera can be designed to try to ensure an exponential decoupled decrease of $\mathbf{e}$. It is given by [6]:

$$\mathbf{v} = -\lambda(\widehat{\mathbf{D}\mathbf{L}_\mathbf{s}})^{+}\mathbf{D}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big), \tag{4}$$

where $\mathbf{v}$ is the virtual camera velocity ($\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ where $\boldsymbol{v}$ is the instantaneous linear velocity) and $\boldsymbol{\omega}$ is the instantaneous angular camera velocity, $\mathbf{L_s}$ is called the interaction matrix and links the motion of the feature in the image to the camera velocity ($\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v}$) and λ is a gain that tunes the convergence rate.

The choice of $\mathbf{s}$ (and thus of $\mathbf{L_s}$) is a key point of this algorithm and is now described in details.

III. Modeling issues for catadioptric cameras

A. Projection models

A unified projection model for central panoramic systems has been proposed by Geyer and Daniilidis [11]. According to this model such cameras can be modeled by a first projection on a sphere with coordinates $(0, 0, \xi)$ followed by a perspective projection on the image plane (see Figure 2). Such a model can be defined using parameter ξ which depends intrinsically of the mirror parameters used in the catadioptric camera.

Assuming that sensor intrinsic parameters are known, point $\mathbf{X} = (X, Y, Z)$ projects in the image plane as $\mathbf{x} = (x, y, 1)$ such that:

$$\mathbf{x} = f(\mathbf{X}) \quad \text{with} \quad \left\{ \begin{array}{rcl} x & = & \frac{X}{Z+\xi\sqrt{X^2+Y^2+Z^2}} \\ y & = & \frac{Y}{Z+\xi\sqrt{X^2+Y^2+Z^2}} \end{array} \right. \tag{5}$$

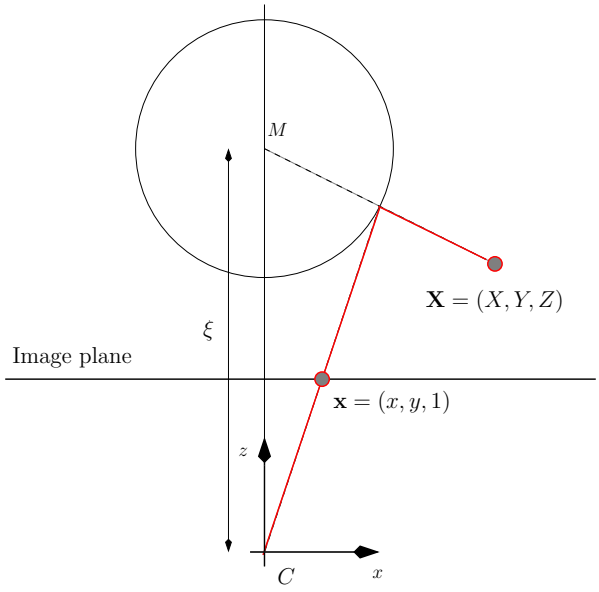

Fig. 2. Generic projection model as defined in [11]

B. Choice of the visual features

In this paper we assume that the 3D model of the object is made of 3D lines. Therefore, following our previous work [6] we consider as visual features the distance between a point $\mathbf{p}$ (extracted from the image, see Section IV) and the projection of this line, (that is an ellipse $\mathbf{e}(\mathbf{r})$) for a given pose .

Therefore in our case vector $\mathbf{s}(\mathbf{r})$ will defined by:

$$\mathbf{s}(\mathbf{r}) = \begin{bmatrix} \vdots \\ s_i(\mathbf{r}) \\ \vdots \end{bmatrix} \quad \text{with} \quad s_i(\mathbf{r}) = d_a(\mathbf{x}, \mathbf{e}(\mathbf{r})) \tag{6}$$

where $d_a(.)$ defines the algebraic distance between point $\mathbf{x}$ and the projection $\mathbf{e}(\mathbf{r})$ of the 3D line (see Section III-E).

In this paper we have chosen to represent a 3D line as the intersection of two planes. We will show how this line projects in the image according to the projection model (5) and how to compute the interaction matrix related to the projection of a 3D line, and then to the considered distance. In [3] a distance between a point projected on the normal of the contour is considered (as in [7] in the perspective case). This leads to a different modeling of the visual feature and of its interaction matrix.

Let us note that other representations of 3D lines exist and can also be considered in this framework. For example Andreff et al. [1] considered a Pluckerian representation of 3D lines and computed the related interaction matrix for classical perspective cameras. This work has been extended in [19] to the case of a catadioptric camera. This Pluckerian representation of 3D line can be easily used as an alternative to the method presented in this paper.

C. Projection of a 3D straight line

A 3D straight line can be represented as the intersection of two 3D planes given by:

$$\begin{array}{rcl} \mathcal{P}_1 : A_1X + B_1Y + C_1(Z-\xi) & = & 0 \\ \mathcal{P}_2 : A_2X + B_2Y + C_2Z + D_2 & = & 0 \end{array} \quad (7)$$

where the 3D parameters are constrained by:

$$\left\{ \begin{array}{l} A_1^2 + B_1^2 + C_1^2 = 1 \\ A_2^2 + B_2^2 + C_2^2 = 1 \\ A_1A_2 + B_1B_2 + C_1C_2 = 0 \end{array} \right. \quad (8)$$

so that the two planes with unit normals $\mathbf{N_1} = (A_1, B_1, C_1)$ and $\mathbf{N_2} = (A_2, B_2, C_2)$ are orthogonal.

According to the projection model defined by (5) the projection of a straight line is nothing but the perspective projection of a circle defined as the intersection between plane $\mathcal{P}_1$ and the sphere $\mathcal{S}$ centered in $(0, 0, \xi)$ with radius 1 (see Figure 2). This 3D circle is characterized by the following system:

$$X^2 + Y^2 + (Z-\xi)^2 = 1 \quad (9)$$

$$A_1X + B_1Y + C_1(Z-\xi) = 0 \quad (10)$$

Considering (9) we directly obtain from [8] the equation of the perspective projection of this circle in the image plane (that is an ellipse) defined by:

$$K_0x^2 + K_1y^2 + 2K_2xy + 2K_3x + 2K_4y + K_5 = 0 \quad (11)$$

with

$$\left\{ \begin{array}{ll} K_0 = 1 + \frac{A_1^2}{C_1^2\xi^2}(\xi^2-1) & K_1 = 1 + \frac{B_1^2}{C_1^2\xi^2}(\xi^2-1) \\ K_2 = \frac{A_1B_1}{C_1^2\xi^2}(\xi^2-1) & K_3 = -\frac{A_1}{C_1\xi^2} \\ K_4 = -\frac{B_1}{C_1\xi^2} & K_5 = -\frac{1}{\xi^2} \end{array} \right. \quad (12)$$

To obtain a minimal representation, let us define $a_i = K_i/K_5, i = 0 \dots 4$, we have then an ellipse defined by:

$$Q(x,y) = a_0x^2 + a_1y^2 + 2a_2xy + 2a_3x + 2a_4y + 1 \quad (13)$$

with

$$\left\{ \begin{array}{ll} a_0 = \frac{A_1^2}{C_1^2}(1-\xi^2) - \xi^2 & a_1 = \frac{B_1^2}{C_1^2}(1-\xi^2) - \xi^2 \\ a_2 = \frac{A_1B_1}{C_1^2}(1-\xi^2) & a_3 = \frac{A_1}{C_1} \\ a_4 = \frac{B_1}{C_1} & \end{array} \right. \quad (14)$$

D. Interaction matrix

Let us now compute the interaction matrix related to parameters a_i. It is possible to rewrite a_0, a_1 and a_2 using a_3 and a_4

$$\left\{ \begin{array}{l} a_0 = a_3^2(1-\xi^2) - \xi^2 \\ a_1 = a_4^2(1-\xi^2) - \xi^2 \\ a_2 = a_3a_4(1-\xi^2) \end{array} \right. \quad (15)$$

Leading to

$$\left\{ \begin{array}{l} \dot{a}_0 = 2a_3(1-\xi^2)\dot{a}_3 \\ \dot{a}_1 = 2a_4(1-\xi^2)\dot{a}_4 \\ \dot{a}_2 = (1-\xi^2)(a_3\dot{a}_4 + a_4\dot{a}_3) \end{array} \right. \quad (16)$$

To obtain the interaction matrix $\mathbf{L_{a_i}}$ for $a_i, i = 0 \dots 4$, we thus need to determine $\mathbf{L_{a_3}}$ $\mathbf{L_{a_4}}$. Considering equation (14) we have

$$\left\{ \begin{array}{l} \dot{a}_3 = \frac{\dot{A}_1}{C_1} - \frac{A_1}{C_1^2}\dot{C}_1 = \frac{\dot{A}_1}{C_1} - a_3\frac{\dot{C}_1}{C_1} \\ \dot{a}_4 = \frac{\dot{B}_1}{C_1} - \frac{B_1}{C_1^2}\dot{C}_1 = \frac{\dot{B}_1}{C_1} - a_4\frac{\dot{C}_1}{C_1} \end{array} \right. \quad (17)$$

It is then necessary to use $\dot{\mathbf{N}}_\mathbf{1} = (\dot{A}_1, \dot{B}_1, \dot{C}_1)$. The time variation $\dot{\mathbf{N}}_\mathbf{1}$ is given by [9][1]:

$$\dot{\mathbf{N}}_\mathbf{1} = -\frac{1}{D_2}\mathbf{N_1}\mathbf{N_2}^\top\boldsymbol{v} - \mathbf{N_1} \times \boldsymbol{\omega} \quad (18)$$

leading to:

$$\begin{array}{rcl} \mathbf{L_{A_1}} & = & \left[\begin{array}{cccccc} -A_1A & -B_1A & -C_1A & 0 & -C_1 & B1 \end{array} \right] \\ \mathbf{L_{B_1}} & = & \left[\begin{array}{cccccc} -A_1B & -B_1B & -C_1B & C_1 & 0 & -A_1 \end{array} \right] \\ \mathbf{L_{C_1}} & = & \left[\begin{array}{cccccc} -A_1C & -B_1C & -C_1C & -B_1 & A_1 & 0 \end{array} \right] \end{array} \quad (19)$$

with

$$A = -\frac{A_2}{D_2}, \quad B = -\frac{B_2}{D_2}, \quad C = -\frac{C_2}{D_2}$$

Plugging (19) in (17) then using (14), we easily obtain:

$$\mathbf{L_{a_3}} = \left[\begin{array}{ccccccc} \alpha a_3 & \alpha a_4 & \alpha & a_3a_4 & -1-a_3^2 & a_4 \end{array} \right] \quad (20)$$

$$\mathbf{L_{a_4}} = \left[\begin{array}{ccccccc} \beta a_3 & \beta a_4 & \beta & 1+a_4^2 & -a_3a_4 & -a_3 \end{array} \right] \quad (21)$$

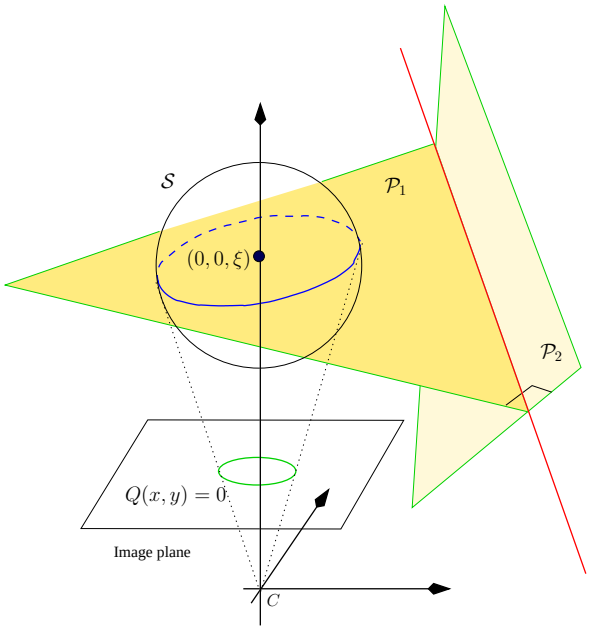

Fig. 3. Algebraic distance between a point and an ellipse

with $\alpha = A - Ca_3$ and $\beta = B - Ca_4$.

Considering (20) and (21) in (16) allows us to deduce the interaction related to a_0, a_1 and a_2 since we have of course:

$$\mathbf{L_{a_0}} = 2a_3(1-\xi^2)\mathbf{L_{a_3}} \quad (22)$$

$$\mathbf{L_{a_1}} = 2a_4(1-\xi^2)\mathbf{L_{a_4}} \quad (23)$$

$$\mathbf{L_{a_2}} = (1-\xi^2)(a_3\mathbf{L_{a_4}} + a_4\mathbf{L_{a_3}}) \quad (24)$$

E. Distance between a point and the projection of a 3D line

There exist various ways to define the distance between a point and an ellipse (ie, here the projection of a 3D line) [20], [10]. In this paper we decide to use the algebraic distance (see Figure 4). Considering a point (x, y), the algebraic distance between this point and an ellipse of equation (13) is given by:

$$d_a = Q(x, y) \quad (25)$$

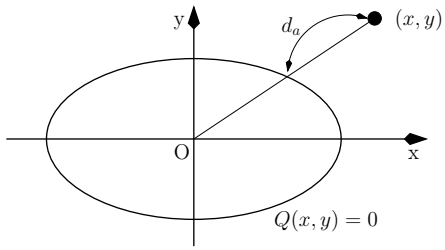

Fig. 4. Algebraic distance between a point and an ellipse

It is then possible to compute the interaction matrix $\mathbf{L_d}$ related to this distance. Considering the time derivative of (25):

$$\dot{d}_a = \dot{a_0}x^2 + \dot{a_1}y^2 + 2\dot{a_2}xy + 2\dot{a_3}x + 2\dot{a_4}y$$

we obtain immediately:

$$\mathbf{L}_{d_a} = \begin{bmatrix} x^2 \\ y^2 \\ 2xy \\ 2x \\ 2y \end{bmatrix}^\top \begin{bmatrix} \mathbf{L_{a_0}} \\ \mathbf{L_{a_1}} \\ \mathbf{L_{a_2}} \\ \mathbf{L_{a_3}} \\ \mathbf{L_{a_4}} \end{bmatrix}$$

IV. Low level image processing

When dealing with low-level image processing, the contours are sampled at a regular distance. At these sample points a 1 dimensional search is performed to the normal of the contour for corresponding edges. An *oriented* gradient mask [4] is used to detect the presence of a similar contour. One of the advantages of this method is that it only searches for edges which are aligned in the same direction as the parent contour. An array of 180 masks is generated off-line which is indexed according to the contour angle. This is therefore implemented with convolution efficiency, and leads to real-time performance.

More precisely, the process consists of searching for the corresponding point $\mathbf{p}_{t+1}$ in image I^{t+1} for each point $\mathbf{p}_t$ (see Figure 5). A 1D search interval $\{\mathbf{Q}_j, j \in [-J, J]\}$ is determined in the direction δ of the normal to the contour.

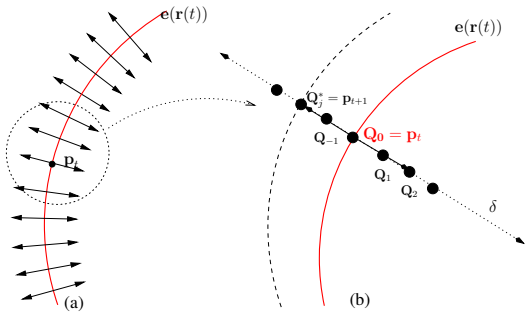

Fig. 5. Determining points position in the next image using the moving edge algorithm: (a) calculating the normal at sample points, (b) sampling along the normal and searching new similar contour.

For each point P_i^t in the list L^t, and for every integer position Q_j, we compute a criterion corresponding to the square root of a log-likelihood ratio ζ^j [4]. The latter is nothing but the absolute sum of the convolution values, computed at p^t and Q_j respectively in images I^t and I^{t+1}, using a pre-determined mask M_δ function of the orientation of the contour. Then the new position $\mathbf{p}_{t+1}$ is given by:

$$Q_j^* = \arg \max_{j \in [-J,J]} \zeta_j \text{ with } \zeta_j = \mid I^t_{\nu(p^t)} * M_\delta + I^{t+1}_{\nu(Q_j)} * M_\delta \mid \quad (26)$$

$\nu(.)$ is the neighborhood of the considered pixel. In this paper the neighborhood is limited to a 7×7 pixel mask. That is a trade-off made between real-time performance and mask size. Likewise there is a trade-off to be made between the search distance and real-time performance while considering the maximum inter-frame movement of the object. Currently, this search distance has been set to 10 pixels.

This low level search produces a list of k points which are used to calculate the distances defined in (25) and used in (6) and (13).

V. Experimental results

The algorithm was tested on real data acquired using two different catadioptric cameras. Two objects were also considered: a box and a set of two plinths. Real images and sensor calibration used to obtain our experimental results are courtesy of the Lasmea. The whole system has been implemented using the ViSP software [17]. Computation time is 100ms for each frame using a 2.6 Ghz PC.

A. Tracking handheld box

In the first experiments a box has been considered. Although the object is quite simple, the object moves very fast which implies very large inter-frame motion as can be seen on the video (see Section V-C). Let us note that despite faces appearance and disappearance (eg, on Figure 6a and 6d or on Figure 7a and 7b), partial occlusions of the object and large contour ambiguity (see Figure 7d) tracking is achieved successfully. The robust minimization based on the M-estimation considered in section II is one of the key of the success of the tracking for these difficult sequences. Without robust estimation (as considered in [3], that if $\mathbf{D} = \mathbf{I}$

in equation (4)) tracking failed almost immediately due to the issues mentioned above.

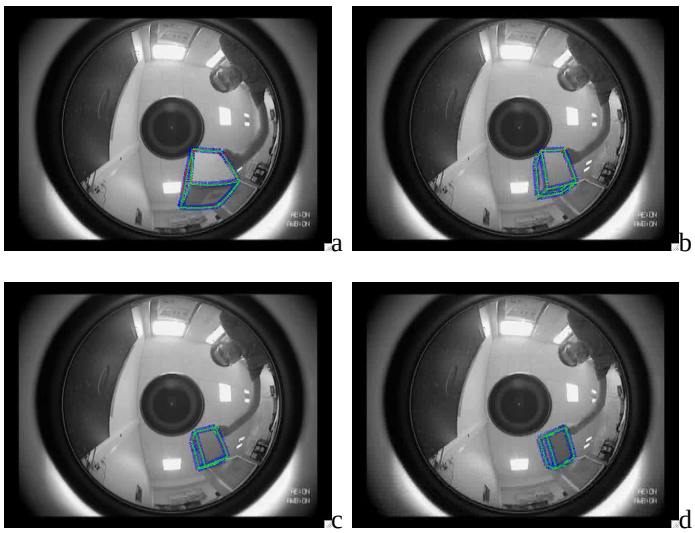

Fig. 6. Tracking a moving box: results on a first sequence

B. Tracking plinths

In this experiment the considered object is composed by two orthogonal plinths and is made with 7 segments. In the image sequence, the sensor, initially static is then handled and moves in various directions (with translations and rotations) and is then finally put back down. Let us note that this is a very long sequence with more than 800 images. The goal is to use such a tracker within a visual servoing system such as [19]. Indeed robust structure tracking is usually one of the key of the success of such navigation systems. In this experiment, despite large and fast motion of the camera tracking is successfully achieved along all the sequence. Figure 8 shows the results of the image processing algorithm. In blue the search line of the moving edge (ME) algorithm is displayed while red points correspond to the point $\mathbf{x}$ (found using the ME) used in equation (6). Yellow parts of ellipses correspond to the projection of the 3D model ($\mathbf{e}(\mathbf{r})$) for the final estimated pose. Figure 9 shows height images of the full sequence with the forward projection of the 3D model.

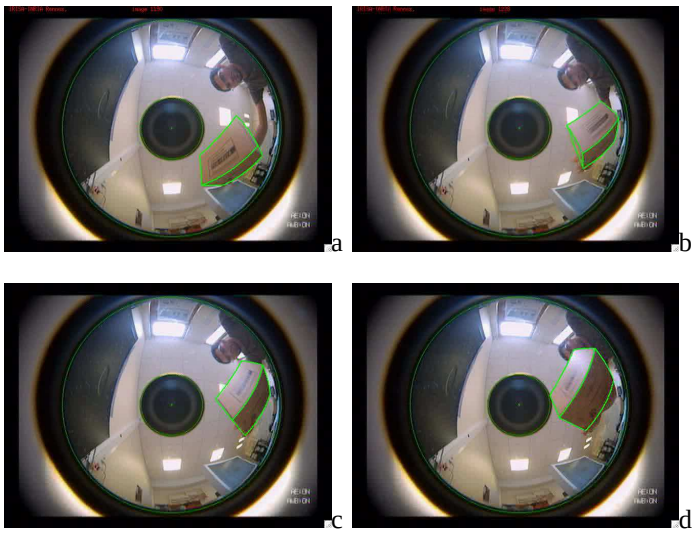

Fig. 7. Tracking a moving box: results on a second sequence

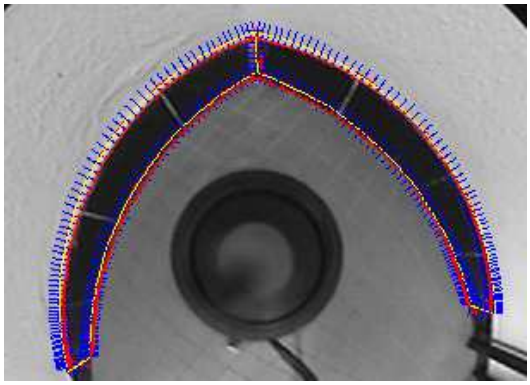

Fig. 8. Tracking plinths: red points correspond to the points $\mathbf{p}$ extracted by the ME algorithm along the normals to the previous 3D model projection (in blue). Forward projection of the model for a given pose is displayed in yellow.

C. Videos

Videos are available from the demo section in the Lagadic web site `http://www.irisa.fr/lagadic`.

VI. Conclusions

We have presented in this paper a 3D model-based algorithm suitable for central catadioptric images. It has proved to be fast and reliable and can therefore be considered in various robotics applications such as visual servoing. Let us note that the parametrization used for the line is an interesting alternative to the one proposed in [19] and can also be used for visual servoing purpose.

Acknowledgments

The authors wish to thank Hicham Hadj-Abdelkader and Maxime Lhuillier from Lasmea who provided respectively the image sequences and the sensor calibration. They also wish to thank Youcef Mezouar and, again, Hicham Hadj-Abdelkader for fruitful discussions.

References

[1] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. *Int. Journal of Robotics Research*, 21(8):669–699, August 2002.

[2] S. Baker and S.K. Nayar. A theory of single-viewpoint catadioptric image formation. *Int. Journal of Computer Vision*, 35(2):175–196, November 1999.

[3] J.P. Barreto, F. Martin, and R. Horaud. Visual servoing/tracking using central catadioptric images. In *Int. Symp. on Experimental Robotics, ISER'02*, pages 863–869, Bombay, India, July 2002.

[4] P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(5):499–511, May 1989.

[5] A.I. Comport, E. Marchand, and F. Chaumette. Robust model-based tracking for robot vision. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, volume 1, pages 692–697, Sendai, Japan, September 2004.

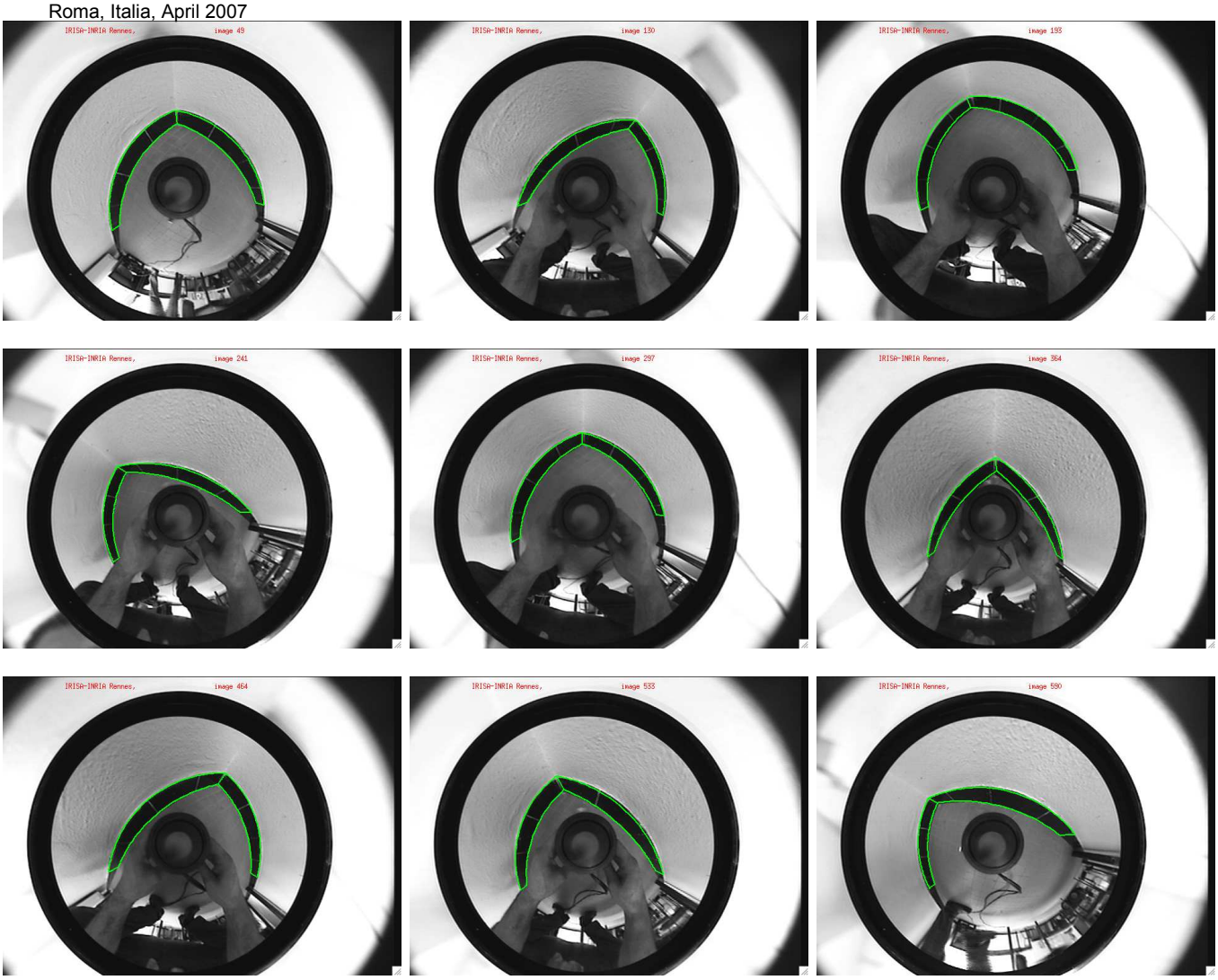

Fig. 9. Tracking a plinth. The plinth is static while the sensor is moving.

[6] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Trans. on Visualization and Computer Graphics*, 12(4):615–628, july 2006.

[7] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.

[8] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[9] B. Espiau and P. Rives. Closed-loop recursive estimation of 3d features for a mobile vision system. In *IEEE Int. Conf. on Robotics and Automation, ICRA'87*, volume 4, pages 1436– 1443, Raleigh, NC, March 1987.

[10] A.W. Fitzgibbon and R.B. Fisher. A buyer's guide to conic fitting. In *British Machine Vision Conference, BMVC'95*, 1995.

[11] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical implications. In *European Conference on Computer Vision ,ECCV'00*, pages 159–179, Dublin, Eire, May 2000.

[12] C. Geyer and K. Daniilidis. Catadioptric projective geometry. *Int. Journal of Computer Vision*, 45(3):223–243, December 2001.

[13] C. Geyer and K. Daniilidis. Mirrors in motion: epipolar geometry and motion estimation. In *IEEE Int. Conf on Computer Vision, ICCV'03*, pages 766–773, Nice, France, 2003.

[14] P.-J. Huber. *Robust Statistics*. Wiler, New York, 1981.

[15] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[16] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[17] E. Marchand, F. Spindler, and F. Chaumette. ViSP for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4):40–52, December 2005.

[18] C. Mei, S. Benhimane, E. Malis, and P. Rives. Homography-based tracking for central catadioptric cameras. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'06*, Beijing, China, October 2006.

[19] Y. Mezouar, H. Haj Abdelkader, P. Martinet, and F. Chaumette. Central catadioptric visual servoing from 3d straight lines. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, volume 1, pages 343–349, Sendai, Japan, September 2004.

[20] P.L. Rosin. Analyzing error of fit functions for ellipses. *Pattern Recognition Letters*, 17(14):1461–1470, December 1996.